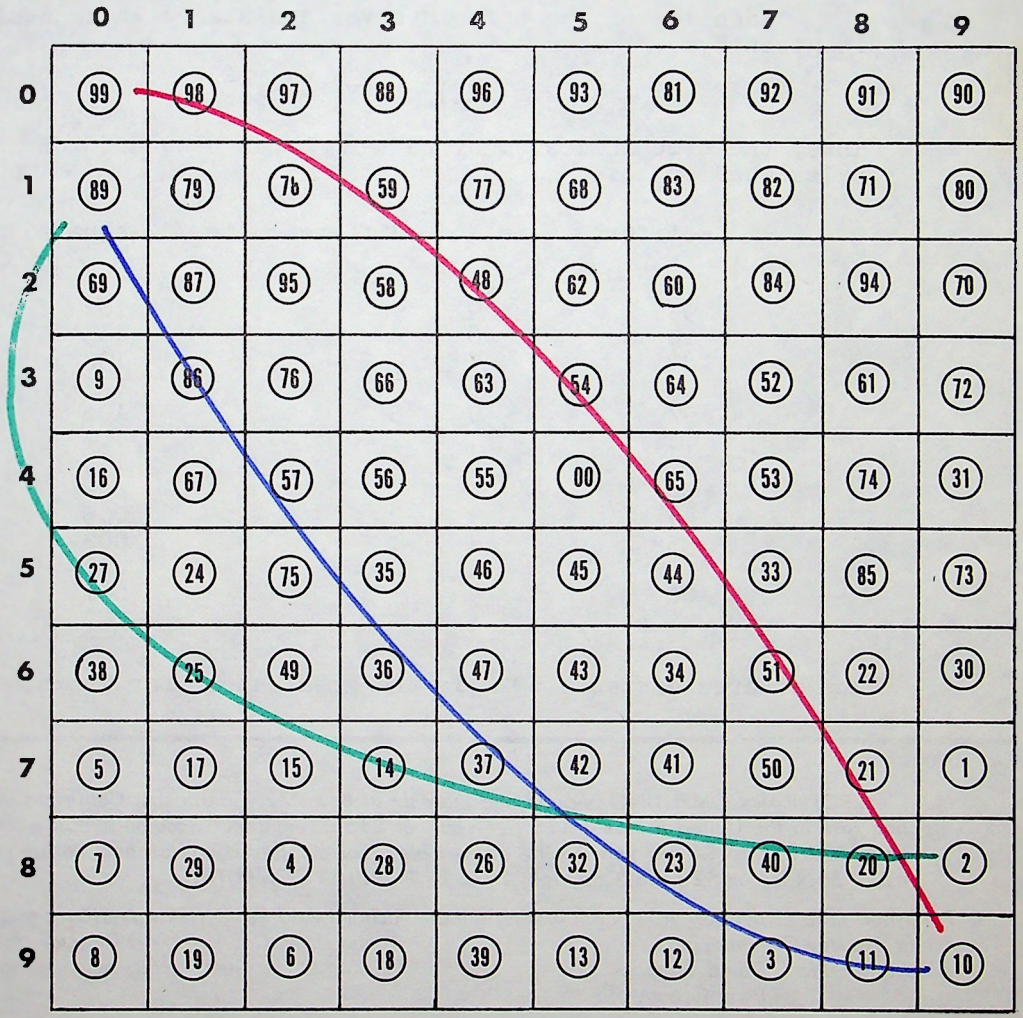
Popular Computing

24

VOL 3 NO 3 MARCH 1975



Q

PC24-2

the Q pattern

The pattern Q on the cover has 100 cells, each of which has an "address" given by its row and column numbers. The 100 cells contain all the 2-digit numbers from 00 through 99. The contents of each cell thus points to another cell. If we enter the pattern at cell 00, its contents point to cell 99, which in turn points to cell 10, whose contents take us to cell 89, and so on. The first three such jumps are shown in color.

For the pattern shown, all 100 cells will be visited, and the trip ends at cell 45 where we return to the starting point.

Each leg of the trip involves a distance whose square is given by:

$$(X_1 - X_2)^2 + (Y_1 - Y_2)^2$$

Thus, the trip indicated by pattern Q has a squared distance calculated by:

	difference in X	difference in Y	squared distance
00			
99	9	9	162
10	8	9	145
89	7	9	130
02	8	7	113
97	9	5	106
..	.	.	...
..	.	.	...
56			
44	1	2	005
55	1	1	002
45	1	0	001
00	4	5	041
			6026

The squared distance 6026 is the greatest so far known.

POPULAR COMPUTING is published monthly at Box 272, Calabasas, California 91302. Subscription rate in the United States is $18 per year, or $15 if remittance accompanies the order. For Canada and Mexico, add $4 per year to the above rates. For all other countries, add $6 per year to the above rates. Back issues $2 each. Copyright 1975 by POPULAR COMPUTING.

Publisher: Fred Gruenberger
Editor: Audrey Gruenberger
Associate Editors: David Babcock, Irwin Greenwald
Contributing editors: Richard Andree, Daniel D. McCracken, William C. McGee
Advertising manager: Ken W. Sims
Art Director: John G. Scott
Business Manager: Ben Moore

PROBLEM 82

Of the 100! possible permutations of the 100 numbers, many do not yield a trip of 100 legs. The probability involved is the same as the one involved in the problem of the Hat Check Girl: a hat check girl in a large theater hands out the customers' hats at random at the end of the show. What is the chance that no customer goes home with his own hat? In the problem at hand, we would be asking for the probability that no number occupies the cell with its own address. This probability approaches 1/e = .36787944... Thus, if the numbers are inserted into the array in random order, the probability is .3679 that some one or more of the numbers will be in a cell having that number for its address.

Is there an arrangement of the 100 numbers that will have a squared-distance greater than 6026?

Semper Erro; Numquam Dubito

PROBLEM 83

Take an integer, N, and form a new number, A, by reversing its digits.

Change both of these numbers to binary notation.

Perform the AND operation (ones wherever both numbers contain a one) on the two binary numbers.

Calculate the ratio of the number of one bits in the product to the total width.

In the first example given below, the result is 6/16 = .375.

Problem: How big can that ratio get?

```
N = 23456      0101101110100000
A = 65432      1111111110011000
Product         1 11 111

N = 1234321    10010110101011001 0001
A = 1234321    10010110101011001 0001
                1  1 11 1 1 11  1   1
```

A Palindromic Problem

Speaking of Languages

ROBERT TEAGUE

This is the third column devoted to the TUTOR language for the PLATO system (see PC21-12 and PC22-13). The concepts of how a lesson is developed and some introduction to the language commands have been given. We now will turn our attention to the TUTOR commands and writing lessons in more detail.

The "display" commands are those which produce the information shown on the screen. The "at" command is used to position the starting point of a display. The screen is divided into 32 lines horizontally and 64 columns vertically. The "at" references a character position in terms of "row-column" within that grid. For example, "at 1510" says to position the next output at line 15, character position 10. The "write" is then used to produce the actual output. Any information given in the tag field of a write will be displayed exactly as given. For instance,

```
at      1204
write   Very Good!
```

will produce, starting at line 12, position 4,

Very Good!

To produce figures, two commands are used. The "circle" command will draw a circle of the given radius with its center at a stated point on the screen. When other figures are needed, the "draw" command is used. It will draw lines between stated beginning and ending points. The command "draw 310;810;810;825;310;825" will produce a right triangle with the 90° angle at position 810 (line 8, character 10). (See the sample lesson for an example of using the draw.)

The judging commands are used to elicit a response from the student and then to determine whether or not it is correct. The "arrow" command displays an arrow on the screen, puts TUTOR in judging state, and awaits an input from the keyboard. The position of the arrow on the screen is given in the same notation as for the "at." The "answer" command indicates the correct response, and the "wrong" gives the incorrect response. More than one "wrong" can be used to cover several possible wrong answers. After the student response to the arrow command has been matched to an "answer" or "wrong," the judging state is terminated and the display instructions following

it will be carried out. In the sample lesson, if the answer is "right triangle" the message "Very Good!" will be displayed. If the answer is "square" the message "A square has four sides." will be given and the student told to try again. The judging state will be reactivated after a "wrong" since it will try to get another, hopefully correct, response from the student. A small loop, therefore, is built between the "wrong" and "arrow" commands until a correct response is obtained.

In the sample lesson, the screen would first be filled with the title LESSON ON FIGURES; then the question "What type of figure is this?" with the picture of a triangle would appear (not necessarily in the order discussed). The "arrow" will appear at position 1930 and the system will await a response. As discussed earlier, the student answers of "right triangle" and "square" will be handled as correct and wrong respectively. A second "wrong" is used to produce a separate and different message if the input is "circle" than if it is "square." With either of the latter two responses, the student will automatically be given another chance to answer correctly.

In our next column, we will conclude our discussion of PLATO with some of the advanced commands and also some discussion of animation and audio response techniques.

Sample TUTOR Lesson

```
unit      figures
at        523
write     LESSON ON FIGURES
at        1010
write     What type of figure is this?
draw      1230;1730,1730;1745,1230;1745
arrow     1930
answer    right triangle
at        2510
write     Very Good!
wrong     square
at        2510
write     A square has four sides.
          Try again.
wrong     circle
at        2510
write     A circle is round.
          Try again.
```

Delay Lines

The problem described here is designed for a class exercise in coding. The procedure is well defined, leading to specific results which are given here as test values. The problem is non-trivial, and lends itself to segmentation via subroutines.

Consider a block of storage of 5 words (or a vector of dimension 5, if you prefer). The block is a push-down list; that is, a new number stored in the first word of the block causes the entire list to be pushed to the right, with the 5th oldest number becoming output at the far right end. For example, if the block contains these numbers:

17	123	88	2345	707
A	A+1	A+2	A+3	A+4

and a new number, 444, is inserted at A, we will then have:

444	17	123	88	2345
A	A+1	A+2	A+3	A+4

and the number 707 will be pushed off the list. The block thus functions as a delay line, so that the numbers emerging as output are delayed five cycles from the current input number.

Consider also the following recursion:

$$X_{n+1} = 7\ X_n \bmod 20123$$

which forms a crude random number generator. If X_0 is taken as 1000, the recursion will yield a stream of numbers starting with:

7000
8754
909
6363
4295
9942
9225 ...

The number 909, for example, is obtained by multiplying 8754 by 7 and finding the remainder on division of 61278 by 20123, which is 909.

We have three delay line blocks, of lengths 5, 7, and 11 words. That is, block A consists of words A through A+4 as shown; block B consists of words B through B+6; block C consists of words addressed C through C+10.

We begin with the three blocks filled with zeros, and the generator initialized to $X_0 = 1000$. The generator (subroutine) is called. The output is tested modulo 3. If the output number is:

0 mod 3: enter the number into line A
1 mod 3: enter the number into line B
2 mod 3: enter the number into line C.

For example, delay line A will have the following contents at various times:

	A	A+1	A+2	A+3	A+4	Output
T	0	0	0	0	0	
T+1	8754	0	0	0	0	
T+2	909	8754	0	0	0	
T+3	6363	909	8754	0	0	
T+4	9942	6363	909	8754	0	
T+5	9225	9942	6363	909	8754	
T+6	4206	9225	9942	6363	909	8754
T+7	16983	4206	9225	9942	6363	909
T+8	8649	16983	4206	9225	9942	6363

The numbers emerging from the delay lines form the output stream. What will be the non-zero values in the output stream? We want the 100th, 200th,...,(100K)th numbers, for which the following results are known:

Position in output stream	Output number	Coming from delay line
100	15705	A
200	879	A
300	19190	C
400	12695	C
500	951	A

and, of the first 200 numbers to emerge, 68 come from line A; 67 from line B, and 66 from line C. The results shown were calculated by Edward Winkenhower, Santa Monica City Schools, programmed in BASIC on an Interdata 70.

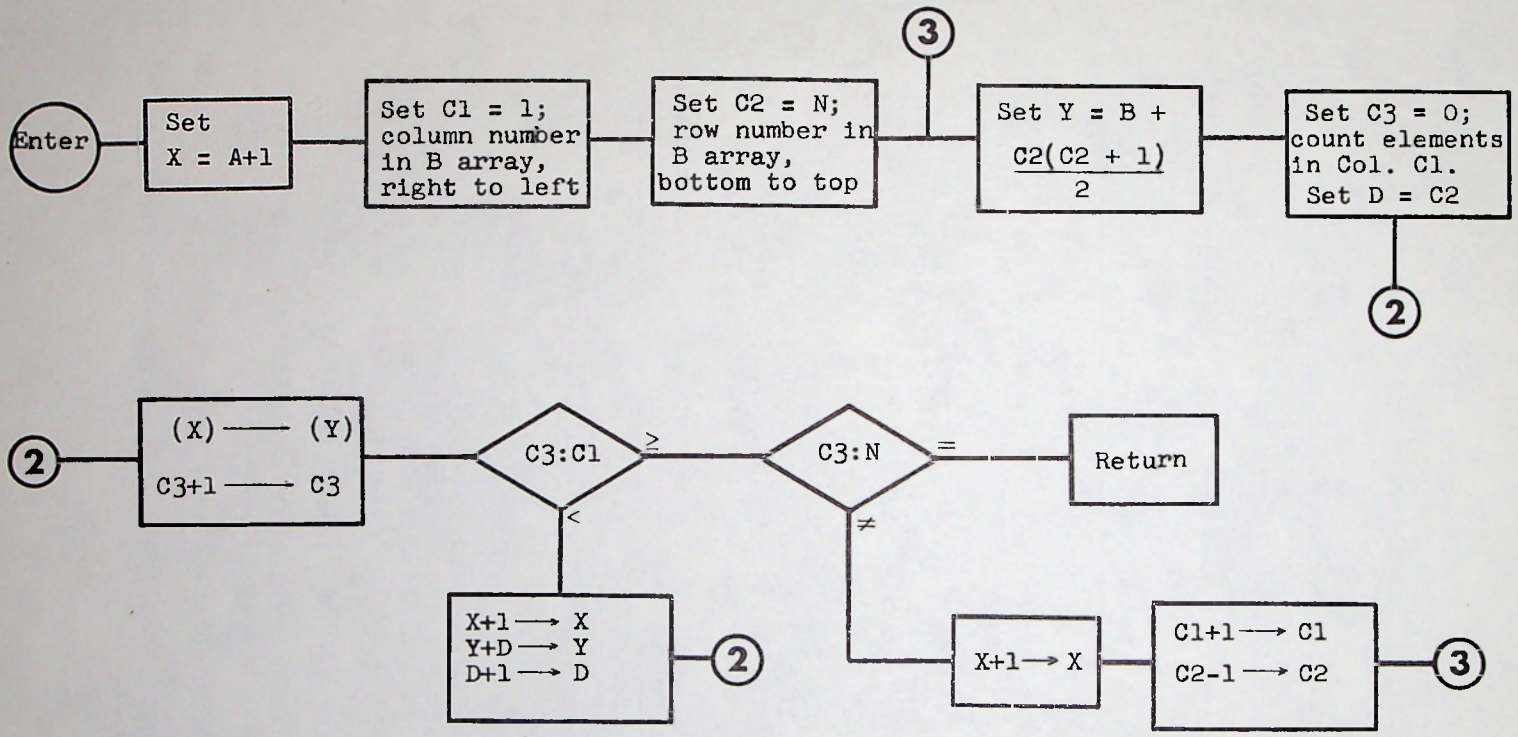
The remaining Problems, then, are these:

(A) Find the numbers that emerge in the (1000K)th positions; that is, the 1000th, 2000th,..., 10000th numbers.

(B) Devise an adequate test procedure for the program.

(C) Generalize the problem for any number of delay lines of any length.

□



A scheme for rotating a triangular array in storage. Block A contains an original array, with N rows (see Figure H in PC23-11). For example, if the array contains 7 rows, the elements occupy cells A+1 through A+28. The numbers in the array are to be moved to block B, rotated 120° clockwise. Again, for N = 7, the number at A+1 is moved to B+28; the number at A+2 is moved to B+21; the number at A+3 is moved to B+27; the number at A+28 will move to B+22; and so on.

Problem Solution

The function listed in this table is the Nth root of N:

Nth Root of N

1	1.0000000	.0000000
2	1.4142136	.2071068
3	1.4422496	.1474165
4	1.4142136	.1035534
5	1.3797297	.0759459
6	1.3480062	.0580010
7	1.3204692	.0457813
8	1.2968396	.0371049
9	1.2765180	.0307242
10	1.2589254	.0258925
20	1.1615863	.0080793
30	1.1200499	.0040017
40	1.0966082	.0024152
50	1.0813827	.0016276
60	1.0706212	.0011770
70	1.0625724	.0008939
80	1.0563033	.0007038
90	1.0512689	.0005697
100	1.0471285	.0004713
150	1.0339684	.0002265
200	1.0268456	.0001342
		1.0700262

This function approaches 1.0 as N increases. If the one's are stripped off, we have a function that approaches zero, but not rapidly enough to converge. The third column, then, is the Nth root less one, divided by N, and that <u>does</u> get smaller fast enough to converge. The sum for the first 200 terms is shown. Theoretical considerations indicate that the true value for

$$\sum_{N \to \infty} \frac{\sqrt[N]{N} - 1}{N} = 1.07057\ldots$$

What is the dividing line between a converging series and a diverging series, both involving terms that approach zero? If a given series is written vertically, and a line is drawn connecting the leading non-zero digits, the shape and direction of that line offer a clue to the distinction. If the line of leading non-zero digits tends toward a straight line with slope different from $-\infty$, the terms of the sequence will be, or will approach,

Converging Series

cr^k where c is a constant and $0 < r < 1$, and the series sum will converge. If the terms decrease faster than this, the line will curve upward, and conversely. All diverging series (with terms approaching zero) will have the line curve downward, with slope approaching $-\infty$. However, the slope may still approach $-\infty$ and permit the series to converge.

In a way, any series of this type that converges can be looked at as a thinned out version of the harmonic series (which is known to diverge). The slope of the line of leading non-zero digits is a measure of the thinning out process. Thus, the series

$$1 + 1/2 + 1/4 + 1/7 + 1/11 + 1/16 + 1/22 + \cdots$$

or the series

$$1 + 1/7 + 1/18 + 1/34 + 1/55 + 1/81 + 1/112 + \cdots$$

perform the thinning fast enough to effect convergence.

Readers are invited to calculate (empirically or analytically) the sums of the series given here, and to comment on the line of distinction between converging and diverging series of the type that can be considered as subsets of the harmonic series. □

N-SERIES 24

Log 24	1.380211241711606022936244587428594389504698508577021
Ln 24	3.178053830347945619646941601297055408873990960903515
$\sqrt{24}$	4.898979485566356196394568149411782783931894961313340
$\sqrt[3]{24}$	2.884499140614816764643276621560219176783738506998701
$\sqrt[5]{24}$	1.888175022589803964328129489455505512452387404230031
$\sqrt[7]{24}$	1.574610106258445653555218820426166832962079919255999
$\sqrt[10]{24}$	1.374108810316637175271392950566869116068732977412152
$\sqrt[100]{24}$	1.032290932124693658268984291930365071840508588200266
e^{24}	26489122129.843472294139162152811882340870198619248530576534206515019619849929321725742606999 6
π^{24}	854273519913.888022186890005793922431963775667067197434536469334334874036686025849392772272 42
$\tan^{-1}$ 24	1.529153747696308195371143626938999229151044035476054

Timewasting

At one time, it was a popular sport among computer people to invent timewasting routines, to consume the largest amount of CPU time while occupying the least number of storage words. For example, the routine

LOCATION	CONTENTS
W	Store zero at word W+4
W+1	Add 1 to word W+4
W+2	Jump on overflow to W+5
W+3	Jump to W+1
W+4	(word used as a counter)
W+5	(continue)

will count 2^M cycles for a word of length M bits. On a machine with a 24-bit word size, where all the instructions given in the routine execute at 4 microseconds each, this short routine will consume 3 minutes, 20 seconds.

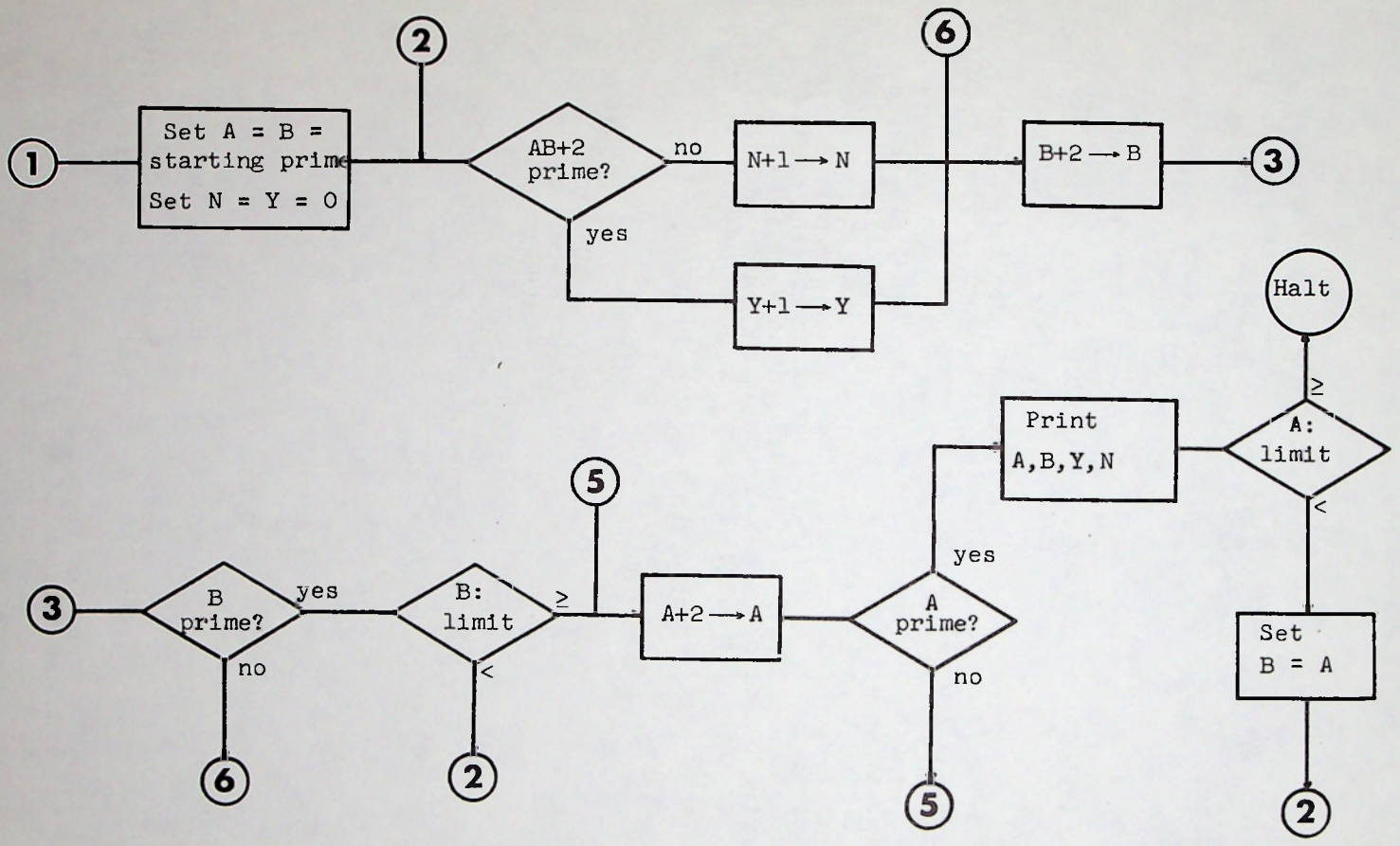
There are (or were) purposes for timewasting routines. As "holding routines," they allowed a programmer, in the days of console debugging, to keep the machine running while the user went for coffee. Or, when magnetic tape operations were more primitive than they are now, such routines were useful to keep the console lights flashing during a tape rewind operation, so that the console operator would not become nervous at seeing the lights freeze. Or, for some jobs that interacted directly with human reactions, a timewasting routine could be used to give the illusion that computation continued while waiting for such reactions.

The following routine:

LOCATION	CONTENTS
W	Store zero at word W+9
W+1	Store zero at word W+10
W+2	Add 1 to word W+10
W+3	Jump on overflow to W+5
W+4	Jump to W+2
W+5	Store zero at word W+10
W+6	Add 1 to word W+9
W+7	Jump on overflow to W+11
W+8	Jump to W+2
W+9	(word used as a counter)
W+10	(word used as a counter)
W+11	(continue)

uses eleven words to count to 2^{2M}. With the same machine parameters as before, this routine will run for 380 years. In both cases, of course, the initial contents of the words used as counters (and/or the amount added) can be adjusted to waste specific amounts of time, from a few microseconds up to the maximum.

The object of the game is to waste the maximum time with the minimum of words. Readers are invited to revive this ancient sport and devise new and clever ways to chew up CPU time. □



A flowchart for one scheme of acquiring information about Problem 79.

To Flowchart or Not to Flowchart

In PC23 this problem was given: What is the probability that the number AB+2 is prime, if A and B are both odd primes? It was suggested that further empirical research was needed. Following is a narrative flowchart of a possible scheme for examining pairs of primes systematically.

1. Establish a starting value for each of two variables, A and B; this value should be a prime. Set two counters, Y and N, to zero.

2. Is the number AB+2 prime? If so, add 1 to Y; if not, add 1 to N.

3. Increase B to the next higher prime (add 2 to B and test for primality until a prime is found).

4. If B has reached a previously established upper bound, increase A to the next prime value. If B has not reached its limit, return to step 2.

5. Print current values of A, B, Y, and N.

6. If A has reached its upper bound, halt; otherwise set B equal to A and return to step 2.

Also given is a conventional flowchart for the same logic. There seems to be an undercurrent of opposition to the use of flowcharts for problem solutions, and the narrative flowchart is one possible alternative.

One indication of the intensity of the prevalent feeling is given by examining any dozen contemporary introductory textbooks. Most of them will include strong statements about the value of flowcharts as a device for aiding logical thinking and for communicating that logic to others. Most of them will then ignore flowcharts for the balance of the book.

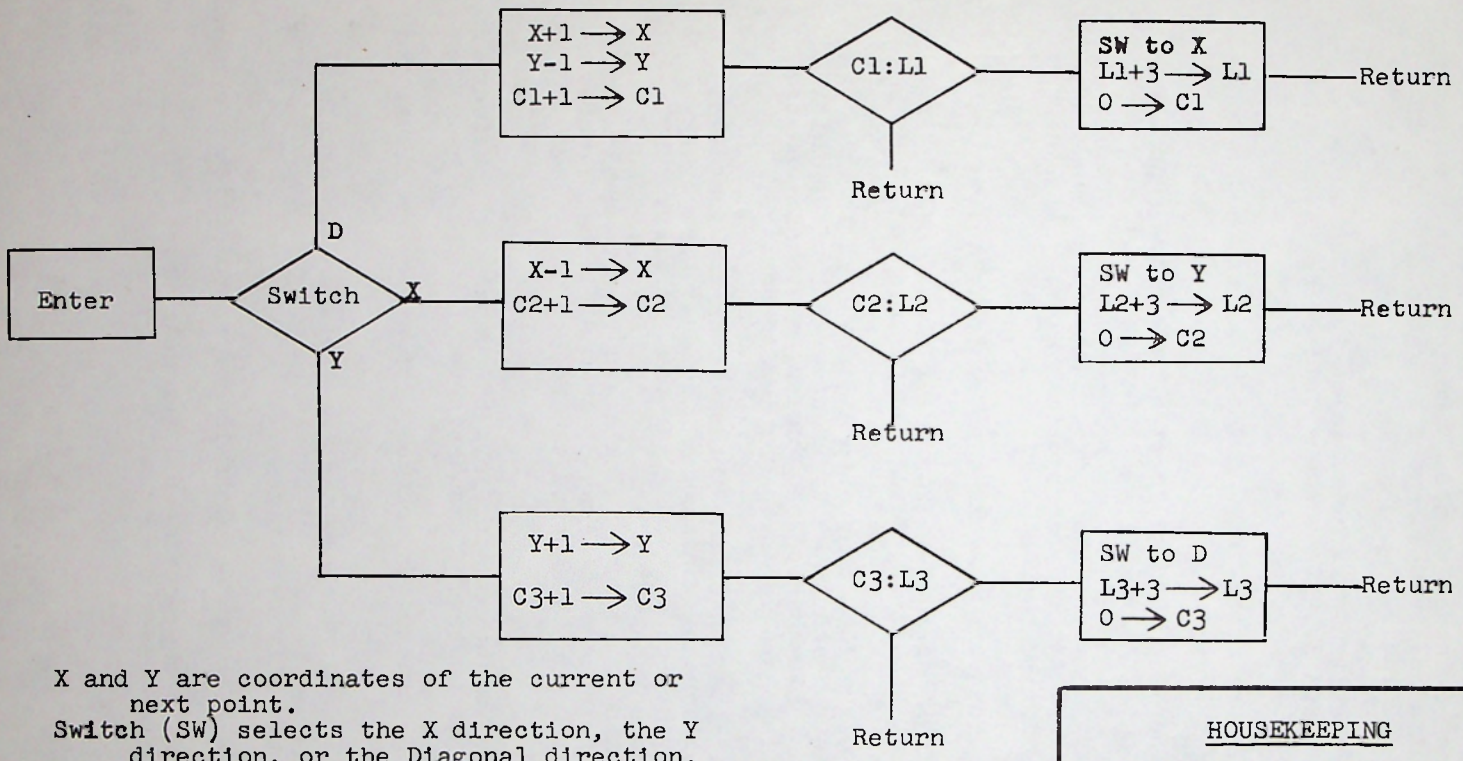
Readers might wish to join the debate. Considering flowcharts (or whatever is substituted for them) as:

(a) A tool to get the logic of a problem situation straightened out;

(b) A tool of communication, between two people or (what is just as likely) between a programmer and himself at two different times;

(c) A tool of documentation of a production program;

--which form is the most useful, or desirable? Or is there something else to use? □



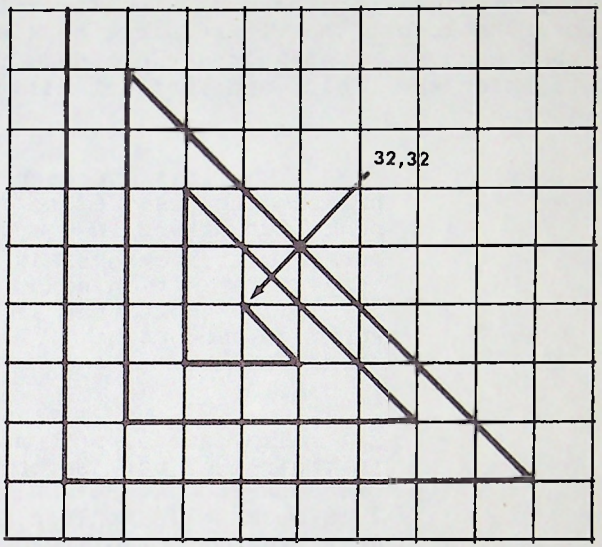
X and Y are coordinates of the current or next point.
Switch (SW) selects the X direction, the Y direction, or the Diagonal direction.
L1, L2, L3 are lengths of the legs in the D, X, or Y directions respectively.
C1, C2, C3 are counters for the distance traversed on a D, X, or Y leg respectively.

HOUSEKEEPING

Set X = Y = 32
Set Switch to D
Set L1 = 1, L2 = 2, L3 = 3
Set C1 = C2 = C3 = 0

Problem Solution

Problem 31 (PC10-9) called for the logic of scanning a lattice in a triangular "spiral," starting at point (32,32) as shown.



The flowchart on the facing page is furnished by Ken Sims, for a subroutine to perform the required task. Since the successive legs of the scan are of lengths 1, 2, 3, 4, 5,..., Mr. Sims suggests that an effective test procedure would be to call the subroutine 15 times and print the output, which should be (30,30); then call it 40 more times, to yield (38,29).

Book Review

COMPUTER CAREERS Planning, Prerequisites, Potential

John Maniotes and James S. Quasney
Hayden Book Company, 1974, 180 pages, $4.95.

A book on vocational advisement in the computing field, which will presumably be read by young people who have little or no knowledge of the field, can go one of three ways:

1. The "This is the field for you" approach. It will stress the pleasant atmosphere, possibilities for rapid advancement, enormity of the field, the wide range of jobs, the rewarding tasks, the prestige involved, and it will minimize grungy things like native talent and hard work (both in preparing to enter the field and in keeping up with it).

2. The "These are the facts" approach. This will be a compendium of statistics about the field, and will probably turn out to have the charm of a government report.

3. A cross between the first two approaches, with the message that the computing field is rewarding and profitable but that it requires work and native talent; that some areas are already crowded; that it's not easy to enter the field, and that not all school programs guarantee a job.

The first approach (and restricted to the area of programming) is the one taken in I. J. Seligsohn's 1967 book Your Career in Computer Programming. Seligsohn's book radiates enthusiasm for the field and is intended to compete for a high school student's attention with dozens of similar books that are touting proctology, or police work, or accounting. It is designed to be an unabashed sales pitch.

The factual approach is taken by Barnett and Davis in their 1967 book Careers in Computer Programming (both the above books are published by Henry Z. Walch, Inc.). This book tells no lies, but would hardly stimulate any enthusiasm for the field.

The latest book, by Maniotes and Quasney, takes a balanced approach and covers the gamut of jobs in the entire DP field. The rewards, financial and otherwise, of work in computing are stated, but qualified with proper warnings of the work involved. The treatment of the advantages and disadvantages of the for-profit DP schools is well done, although the bright side of using these schools is not brought out completely; the reader should refer, for example, to

"Private EDP Schools: The Positive View"
by Leon Cooper, in the book
The EDP People Problem
Data Processing Digest, 1971

(Incidentally, the address given for Data Processing Digest in this book is four years out of date--one of many such minor errors.)

The book contains extensive reference lists for each topic covered, and ten appendices (a book list, a film list, a list of computing societies, sample aptitude tests, etc.

Profs. Maniotes and Quasney disagree with the conclusions reached in the article "The Future of Programmers" (PC19-13)--and they may be right. In any event, they have produced a book that is much needed in vocational counseling. □

Printed by Argold Press Inc.
16921 Ventura Boulevard, Encino